**探索粒子群優化算法在解決實參數最佳化問題中的效能表現**

|  |
| --- |
| 黃梓濤 |
| 國立中山大學 |
|  |

# 摘要

Particle Swarm Optimization (PSO) 是一種源自於觀察鳥群覓食行為的啟發式演算法。鳥群在尋找食物時，雖然能感知自身與食物之間的距離，卻不知道食物的確切位置。因此，鳥群會同時朝不同方向探索食物，並在休息時分享彼此的資訊。

PSO 演算法的運作方式模擬了鳥群的行為。每隻粒子會根據其上次的飛行方向（速度 V(t)）、自身曾到達的最佳位置（個體最佳解 pBest）以及整個群體已達到的最佳位置（全域最佳解 gBest），來決定下一步的移動方向。粒子之間會透過共享資訊的方式，漸漸靠近全域最佳解，模擬了鳥群逐漸靠近食物的行為。

本實驗旨在探討 PSO 演算法在解決實參數單目標最佳化問題上的表現。我們選擇了六種不同的測試函數，包括 Ackley、Rastrigin、HappyCat、Rosenbrock、Zakharov 和 Michalewicz，並比較了不同維度和粒子數量對演算法性能的影響。透過實驗分析， PSO 演算法在不同情境下的優缺點，以及其在實參數最佳化問題中的應用潛力，並深入了解 PSO 演算法的特性

**關鍵詞：啟發式演算法、PSO、粒子群最佳化。**

## 簡介

在實參數單目標最佳化問題中，有效解決高維度、非線性和複雜的優化問題是一個具挑戰性的問題。為了應對這一挑戰，實驗採用了Particle Swarm Optimization (PSO)演算法，該演算法模擬了鳥群尋找食物的行為，具有全局搜尋和快速收斂的優勢。

我旨在探討PSO演算法在解決實參數單目標最佳化問題上的效能和應用。我們將使用六種不同的測試函數，包括Ackley、Rastrigin、HappyCat、Rosenbrock、Zakharov和Michalewicz，這些函數具有不同的特性和局部最佳解，可以有效地評估PSO演算法的性能。

我們將通過對PSO演算法的實驗設計來解決問題。我們將使用不同維度和粒子數量的參數組合，並通過比較不同情境下的演算法性能來評估其效能。

最終，我期望透過本研究的實驗結果，提供對PSO演算法在實參數單目標最佳化問題中的應用和優化方向的有更深入理解。

## 相關研究

* 1. **粒子群最佳化演算法**

粒子群最佳化（Particle Swarm Optimization，PSO）是一種啟發式演算法，源自對鳥群覓食行為的觀察。PSO透過模擬鳥群在空間中的移動，以尋找最佳解。算法中每個解被視為一個粒子，而這些粒子在最佳化問題的解空間中移動位置。在 PSO 中，每個粒子根據個體自己的慣性和粒子自身的經驗，以及群體的經驗，來調整自己的移動方向和速度，在不同方向探索到更好的解。

1. 初始化粒子位置和速度：

* 隨機生成一個粒子群，每個粒子都有一個初始位置和速度。
* 粒子的初始位在解空間範圍內隨機生成的。
* 粒子的初始速度在限定範圍內隨機生成的。

2. 直到迭代結束前歷遍每個粒子：

* 更新粒子速度

第 *i* 個粒子的新速度 w是慣性權重 ​

和是加速係數 是該粒子的個體最優解 是整個粒子群的全局最優解 和 [0,1]的隨機數。

* 更新粒子位置

是第 *i* 個粒子的新位置。

* 更新個體最優解

比較其新位置的目標函數值和其個體最優解的目標函數值，如果新位置更好，則將其設置為新的個體最優解。

* 更新全局最優解

比較所有粒子的個體最優解，找出最優的解作為全局最優解。

1. **實驗設計**

在這個實驗中，我們將使用Particle Swarm Optimization (PSO)演算法來解決實參數單目標最佳化問題。我們將針對六種不同的測試函數（Ackley、Rastrigin、HappyCat、Rosenbrock、Zakharov 和 Michalewicz）進行實驗，並分別評估其在2維、10維和30維空間中的性能。

我們將使用線性遞減慣性權重（linear decrease）的方式來調整粒子的速度更新，其中慣性權重

的計算方式為

，其中 *wMax* 和 *wMin* 分別表示慣性權重的最大值和最小值， *t* 是當前迭代次數，而 *maxIter*  則是設定的最大迭代次數。

在PSO演算法中，我們將設定加速係數 *c1* 和 *c2* 都為1.5，然後分別使用50、100和200個粒子來構建粒子群，以觀察不同粒子數量對演算法性能的影響。

對於每個測試函數，我們將分別計算其平均適應度（Average Fitness）、最差適應度（Worst Fitness）和最佳適應度（Best Fitness），以評估PSO演算法在不同維度和不同粒子數量下的表現。這些評估指標將幫助我們了解PSO演算法的搜索能力和收斂速度。

1. **實驗結果**

**表1. Ackley在粒子數 50實驗結果**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 平均適應度 | 最差適應度 | 最佳適應度 |
| **Ackley 2D** | **5.30E-03** | **1.75E-02** | **3.21E-04** |
| **Ackley 10D** | **3.20E+00** | **6.92E+00** | **2.96E-02** |
| **Ackley 30D** | **5.78E+00** | **7.94E+00** | **3.71E+00** |

以Ackley實驗結果為例適應度表現並不理想，10維 和 30維 和最佳解0E0差距很大，可見粒子群在高維度的問題上全局探索能力上的不足，容易陷入局部最佳解。最佳和最差適應度之間的差距表明了粒子群在探索解空間時遇到的困難。這種差距的擴大通常表明粒子群很可能陷入了局部最優解，而無法有效地搜索到更好的解。

**A graph with numbers and lines

Description automatically generated**

**圖一 Ackley 2維空間的收歛**

**A graph with a red line

Description automatically generated**

**圖二 Ackley 10維空間的收歛**

**A graph with a green line

Description automatically generated**

**圖三 Ackley 30維空間的收歛**

在圖一至圖三中，Ackley在不同維度上收斂的表現能看出，雖然越高維度的測試函數有著再多的最大迭代次數，但是每個維度都在相似的次數中完成收斂，可見在高維度測試函數中更容易陷入局部最佳解，提高迭代次數無助得到更佳的結果，應調整其他參數增加算法的探索能力。

而無法有效地搜索到更好的解。

A graph of a number of people with different numbers

Description automatically generated with medium confidence

**圖四 PSO演算法不同個體數對平均適應度的影響**

圖四中，個體數量對算法性能的影響在不同維度下呈現出不同的特點。在2維Ackley函數的情況下，增加個體數量對適應度的改善影響不大。然而，在高維度的情況下，增加個體數量對算法性能的改善效果更加顯著。

對於10維和30維Ackley函數，隨著個體數量的增加，適應度呈現明顯的下降趨勢。這可能是因為高維度空間中搜索更加困難，需要更多的個體數量才能獲得更好的結果。

同時，隨著維度的增加，測試函數的挑戰性也增加。在10維和30維Ackley函數的實驗結果中，這一趨勢得到了體現。在高維度下，算法更難找到全局最優解，需要更多的個體數量和更長的搜索時間。

A bar graph with different colored bars

Description automatically generated

**圖五 對比不同測試函數表現**

著HappyCat函數是一個非凸函數，整體上是一個相對較為平滑的函數。這使得粒子群優化算法較容易找到較好的解。

相反，Rosenbrock函數是一個高度非線性的函數，具有一個窄而深的碗形結構，其表面有許多平坦的區域和一個突出的最小值。這使得粒子群優化算法很難有效地探索解空間，容易陷入局部最佳解，導致平均適應度表現較差。

## 結論

根據Ackley實驗結果，我們觀察到在高維度情況下，粒子群在全局探索能力上存在不足，容易陷入局部最佳解。這一現象在10維和30維Ackley函數中尤其顯著，最佳和最差適應度之間的差距擴大，表明了粒子群在探索解空間時遇到的困難。

PSO在不同維度下的收斂情況。儘管在高維度情況下迭代次數增加，但收斂的效果並不明顯。這能看出在高維度測試函數中更容易陷入局部最佳解，增加迭代次數並不能有效改善結果。

圖四展示了個體數量對算法性能的影響。在2維Ackley函數中，增加個體數量對適應度改善影響不大，但在高維度情況下，增加個體數量可以明顯改善算法性能。然而，對於10維和30維Ackley函數，隨著個體數量增加，適應度呈現下降趨勢，這可能是因為高維度空間中搜索困難。

高維度的測試函數增加了算法的複雜性和難度，粒子群容易受到局部最佳解的影響。因此，需要繼續調整算法參數或採取其他策略以提高全局搜索能力，以更好地應對高維度的優化問題。